

Лабораторная работа №2

Создание контейнерного приложения в IBM Cloud Private.

Требования:

Наличие доступа к кластеру IBM Cloud Private .

Установленные на локальной машине – Git, Docker, текстовый редактор.

Подключение по VPN к IBM Client Center Moscow.

Успешно выполненная лабораторная работа №1

Этап 1. Подключение к кластеру IBM Cloud Private.

В данной лабораторной работе будет использован демо-кластер IBM Cloud Private, расположенный в IBM Client Center Moscow.

Для подключения к кластеру необходимо выполнить следующие шаги:

- Подключиться по VPN
- Перейти по ссылке - <https://172.17.207.8:8443>
- Логин/пароль – admin/admin
- После успешного входа вы увидите следующий экран (рис.1).

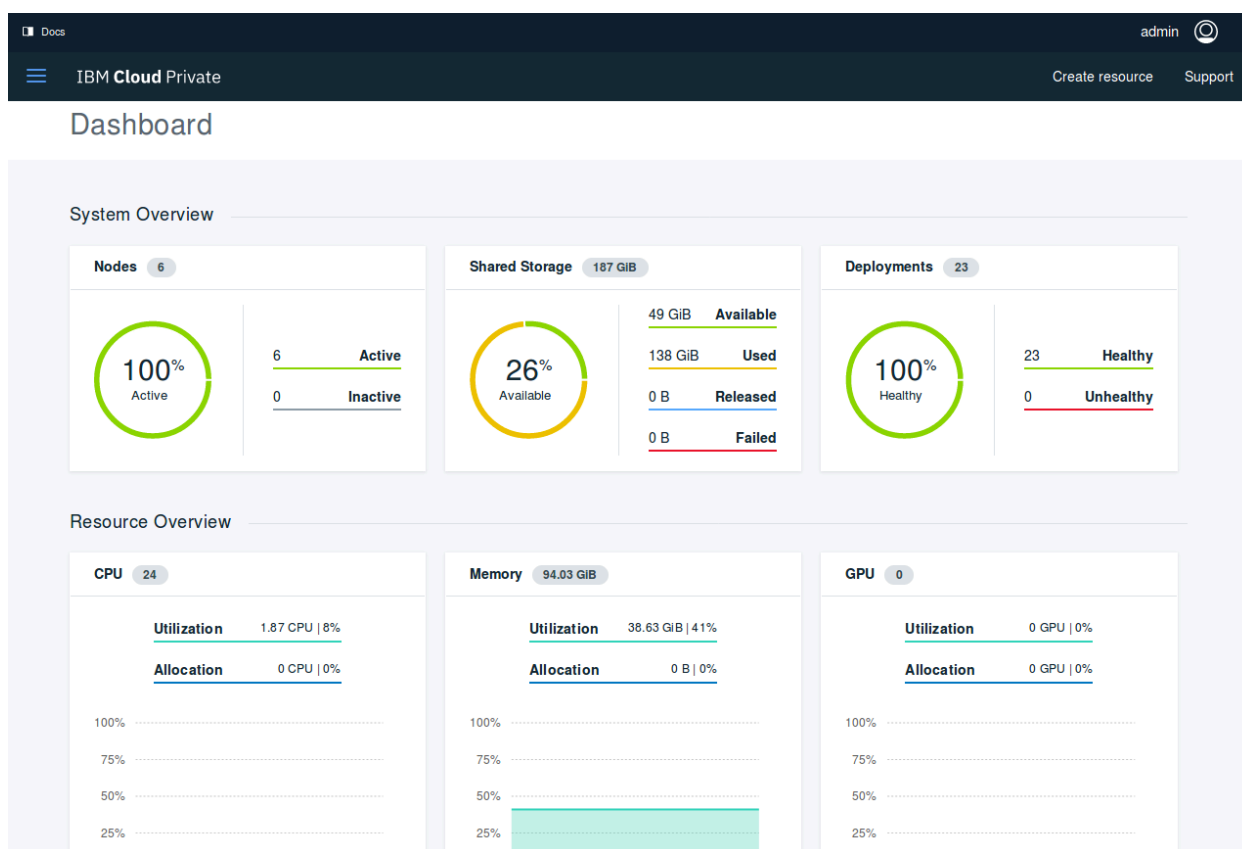



рисунок 1

- Необходимо раскрыть меню, нажав в правом верхнем углу на значок  и в выпадающем списке выбрать опцию Configure Client.
- Из открывшегося окна (рис. 2) необходимо скопировать настройки для kubectl

Configure kubectl

Before you run commands in the kubectl command line interface for this cluster, you must configure the client.

Prerequisites:

Install the kubectl CLI: kubectl

To configure the CLI, paste the displayed configuration commands into your terminal window and run them.:

```
kubectl config set-cluster mycluster.icp --server=https://172.17.207.8:8001
kubectl config set-context mycluster.icp-context --cluster=mycluster.icp
kubectl config set-credentials mycluster.icp-user --token=eyJhbGciOiJSUz
kubectl config set-context mycluster.icp-context --user=mycluster.icp-user
kubectl config use-context mycluster.icp-context
```

рисунок 2

- Необходимо открыть терминал и вставить ранее скопированные значения.
- Проверить подключение можно, например, командой `kubectl get no`
- На этом этап 1 завершен

Этап 2. Подключение к Docker реестру в IBM Cloud Private

- Необходимо прописать имя кластера в файл `/etc/hosts`
`echo 172.17.207.8 mycluster.icp | sudo tee -a /etc/hosts`
- Создать директорию для хранения сертификата
`sudo mkdir -p /etc/docker/certs.d/mycluster.icp:8500/`
- Скопировать с мастер хоста файл с сертификатом (пароль – passwd)
`sudo scp root@172.17.207.2:/etc/docker/certs.d/mycluster.icp:8500/ca.crt /etc/docker/certs.d/mycluster.icp:8500/`
- Залогинится в реестр
`docker login mycluster.icp:8500`

Этап 3. Запуск приложения в IBM Cloud Private.

- Создайте новое пространство имен в кластере
`kubectl create namespace <my_namespace>`

- Создайте метку для локального образа
`docker tag demoapp mycluster.icp:8500/<my_namespace>/demoapp:1`
- Загрузите локальный образ в реестр
`docker push mycluster.icp:8500/<my_namespace>/demoapp:1`
- Перейдите в каталог `demoapp`
- Скопируйте файл `demoapp.yml` в `demoapp-icp.yml`
- Отредактируйте файл `demoapp-icp.yml`, изменив в нем ссылку на локальные реестр

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: demoapp
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: demoapp
    spec:
      containers:
        - name: node
          image: mycluster.icp:8500/<my_namespace>/demoapp:1
          ports:
            - containerPort: 3000
```

- Запустите установку
- `kubectl -n <my_namespace> create -f demoapp-icp.yml`
- Проверьте, что установка прошла успешно
`kubectl -n <my_namespace> get deploy`

`kubectl get po` (рис. 9)

- Получите список сервисов
`kubectl -n <my_namespace> get svc` (рис. 3)

```
ibm@containers-vm:~/demoapp$ kubectl -n my-namespace get svc
NAME          CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
demoapp       10.0.0.245   <pending>     3000:32186/TCP   22m
mongo         10.0.0.104   <none>        27017/TCP        22m
mongo-express 10.0.0.131   <pending>     8081:31096/TCP   22m
```

рисунок 3

- Сервисы доступны адресу 172.17.207.10 и соответствующему порту, например, в данном случае demoapp доступен по адресу <http://172.17.207.10:32186>, а mongo-express по адресу <http://172.17.207.10:31096>
- На этом третий этап и вся лабораторная работа успешно завершены.